

# Load Balacing & Session Clustering

📅 날짜	@2020년 12월 21일 → 2020년 12월 22일
🏷 태그	

→ **nginx + tomcat (docker) load balancing**

## 1. nginx load balancing setting

→ upstream : 각 포트별 로드밸런싱에 사용될 포트를 지정

```
http {  
    upstream http_stream {  
        server 127.0.0.1:8080;  
        server 127.0.0.1:8081;  
    }  
  
    upstream https_stream {  
        server 127.0.0.1:8443;  
        server 127.0.0.1:8444;  
    }  
}
```

→ HTTP setting

```

server {
    listen      80;
    listen      [::]:80;
    server_name _;
    root        /usr/share/nginx/html;

    # Load configuration files for the default server block.
    #include /etc/nginx/default.d/*.conf;

    location / {
        #proxy_pass http://localhost:8080;
        proxy_pass http://http_stream;
        proxy_set_header Host $http_host;
    }

    error_page 404 /404.html;
        location = /40x.html {
    }

    error_page 500 502 503 504 /50x.html;
        location = /50x.html {
    }
}

```

→ HTTPS setting

```

server {
    listen      443 ssl http2;
    listen      [::]:443 ssl http2;
    server_name _;
    root        /usr/share/nginx/html;

    #rewrite ^ https://awslh.xl.co.kr:8443/login.do;

    ssl_certificate "/etc/pki/certs/star_xl_co_kr_cert.pem";
    ssl_certificate_key "/etc/pki/private/star_xl_co_kr_key.pem";
    ssl_session_cache shared:SSL:1m;
    ssl_session_timeout 15m;
    ssl_ciphers PROFILE=TLS13:CHACHA20-POLY1305-SHA256:TLS13-AES-128-GCM-SHA256:TLS13-AES-256-GCM-SHA384:ECDSA:!COMPLEMENTOFDEFAULT;
    ssl_password_file "/etc/pki/private/star_xl_co_kr_key.pass";
    ssl_prefer_server_ciphers off;

    # Load configuration files for the default server block.
    #include /etc/nginx/default.d/*.conf;

    location / {
        #proxy_pass https://localhost:8443;
        proxy_pass https://https_stream;
        proxy_set_header Host $http_host;
    }
}

```

## 2. 각 tomcat별 로그 확인

testOne, testTwo URL을 80포트에 접속하게 한 뒤 로그가 각각 어디에 찍히는지!

nginx 80번 포트를 통해 URL을 입력했지만 docker로 올린 각 tomcat에 로그가 찍히는 것을 확인할 수 있다.

tomcat #1

```
172.17.0.1 - - [21/Dec/2020:09:07:51 +0000] "GET /login.do_testOne HTTP/1.0" 404 732
```

tomcat #2

```
172.17.0.1 - - [21/Dec/2020:09:07:55 +0000] "GET /login.do_testTwo HTTP/1.0" 404 732
```

### → Session Clustering

<https://jstol.github.io/java/2017/09/15/tomcat-clustering/>

위 링크를 보고 아주 간단하게 세션 클러스터링을 설정할 줄 알았지만 역시나 실패

→

8443, 8444 포트로 접속했을 경우 세션 ID 동일한 것 확인

### → <distributed/>

해당 구문을 web.xml에 추가하게 되면 setAttribute 부분에서 에러가 발생한다

```
java.lang.IllegalArgumentException: setAttribute: 직렬화할 수 없는 속성 [loginStatus]  
    at org.apache.catalina.session.StandardSession.setAttribute(StandardSession.java:1416)  
    at org.apache.catalina.session.StandardSession.setAttribute(StandardSession.java:1376)
```

<http://blog.naver.com/estern/110011499088>

톰캣은 재구동 시에 세션을 유지하도록 설정되었고, 해당 설정은 default로 적용된다. 이 설정은 톰캣 종료 시 세션 정보를 직렬화하여 시스템 파일에 저장하고 다시 로드하는 방식으로 작동한다. 따라서 세션에 담기는 모든 attribute는 직렬화하여 저장된다.

<https://tomcat.apache.org/tomcat-8.0-doc/cluster-howto.html>

위의 링크에 따라 진행한 두가지 사항 외에 반드시 필요한 것이 하나 더 있었다.

1. Uncomment the **Cluster** element in server.xml
2. Make sure youre web.xml has the <distributed/> element
3. **All your session attributes must implement java.io.Serializable**

따라서 MemberVO(loginStatus), RSAUtil(rsaPrivateKey) 에 implement를 진행하였다.

## Cluster Basics

To run session replication in your Tomcat 8 container, the following steps should be completed:

- All your session attributes must implement `java.io.Serializable`
- Uncomment the `Cluster` element in `server.xml`
- If you have defined custom cluster valves, make sure you have the `ReplicationValve` defined as well under the `Cluster` element in `server.xml`
- If your Tomcat instances are running on the same machine, make sure the `Receiver.port` attribute is unique for each instance, in most cases Tomcat is smart enough to resolve this on its own by autodetecting available ports in the range 4000-4100
- Make sure your `web.xml` has the `<distributable/>` element
- If you are using `mod_jk`, make sure that `jvmRoute` attribute is set at your Engine `<Engine name="Catalina" jvmRoute="node01">` and that the `jvmRoute` attribute value matches your worker name in `workers.properties`
- Make sure that all nodes have the same time and sync with NTP service!
- Make sure that your loadbalancer is configured for sticky session mode.

## → 각 톰캣을 접속 시 session id 일치 여부 확인

Name	Value	Domain	Path	Expires / ...	Size	HttpOnly	Secure	SameSite	Priority
BMR	s=1608599768641&lr=https%3A%2F%2Fm.blog.naver.com...	.naver.com	/	Session	115				Medium
_gid	GA1.3.145876107.1608600415	.x1.co.kr	/	2020-12-...	30				Medium
JSESSIONID	A58EC80D2D3C8FDC05D698000C4C6EA5	aws1h.x1...	/	Session	42	✓			Medium

Name	Value	Domain	Path	Expires / ...	Size	HttpOnly	Secure	SameSite	Priority
BMR	s=1608599768641&lr=https%3A%2F%2Fm.blog.naver.com...	.naver.com	/	Session	115				Medium
_gid	GA1.3.145876107.1608600415	.x1.co.kr	/	2020-12-...	30				Medium
JSESSIONID	A58EC80D2D3C8FDC05D698000C4C6EA5	aws1h.x1....	/	Session	42	✓			Medium

브라우저가 8443, 8444 포트로 각각 다른 톰캣에 접속했음에도 불구하고 Session ID가 일치하는 것을 볼 수 있다.

또한 nginx를 통해 로그인을 진행한 후 8443, 8444 포트에 따로 접속하는 경우 로그인 상태가 그대로 유지되는 것을 확인할 수 있다.